CSS ID selector

CSS ID with leading hash.

```
#[a-zA-Z_][a-zA-Z0-9_-]*
```

Token-by-token breakdown

Token	Meaning
`#`	literal text “#”
`[a-zA-Z_]`	any of: lowercase letters, uppercase letters, “_”
`[a-zA-Z0-9_-]*`	zero or more: any of: lowercase letters, uppercase letters, digits, “_”, “-”

Quick usage in 7 languages

JavaScript

```
const re = /#[a-zA-Z_][a-zA-Z0-9_-]*/;
re.test(value);
```

Python

```
import re
re.match(r"#[a-zA-Z_][a-zA-Z0-9_-]*", value)
```

Java

```
Pattern.compile("#[a-zA-Z_][a-zA-Z0-9_-]*").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"#[a-zA-Z_][a-zA-Z0-9_-]*");
```

Go

```
regexp.MustCompile(`#[a-zA-Z_][a-zA-Z0-9_-]*`).MatchString(value)
```

Ruby

```
/#[a-zA-Z_][a-zA-Z0-9_-]*/.match?(value)
```

PHP

```
preg_match('~#[a-zA-Z_][a-zA-Z0-9_-]*~', $value);
```

Common pitfalls

- **Not anchored.** Without ^ and $ this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u flag.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the css id selector against a source of truth (database, API, or checksum) where it matters.